

**IN THE CLAIMS:**

Claims 1, 6, and 9 are amended herein. Please cancel claims 12-19 without prejudice or disclaimer. All pending claims and their present status are produced below.

1. (Currently amended) A multithreaded computer based system for enabling a command in a first thread to access data in a second thread comprising:  
an embedded pipelined processor capable of having a first program thread and a second program thread in an execution pipeline, said first program thread comprising a first set of instructions, said second program thread comprising a second set of instructions, said embedded processor comprising:  
a fetch unit for fetching an instruction from program memory;  
a decode unit for decoding said fetched instruction;  
an execution unit for executing said decoded instruction; and  
a write back unit for writing the results of said executed instruction to an identified storage location;  
a first set of data storage devices capable of storing a first state of said embedded processor, wherein said first state is the state of the embedded processor during the execution of the first program thread;  
a second set of data storage devices capable of storing a second state of said embedded processor, wherein said second state is the state of the embedded processor during the execution of the second program thread;  
wherein at least said first set of data storage devices includes a first control status register for identifying a first target set of data storage devices from which a first source operand of a fetched instruction is to be retrieved and for

identifying a second target set of data storage devices to which a first result of an executed instruction is to be stored, wherein at least one of said first or said second target set of data storage devices is not included in the first set of data storage devices;

a thread scheduler for identifying which of said program threads said embedded processor executes; and

an instruction set including an instruction that overwrites the first control status register when instructions associated with the first set of data storage devices are executed and overwrites ~~the~~ a second control status register when instructions associated with the second set of data storage devices are executed;

wherein said processor switches between said first and second state in a time period between the end of the execution of a first program instruction in the first thread and the beginning of the execution of a second program instruction in the second thread;

wherein said processor switches between said first and second states by changing a state selection register.

2. (Previously presented) The multithreaded computer based system of claim 1, wherein the embedded pipelined processor further includes a peripheral block.

3. (Previously presented) The multithreaded computer based system of claim 2, wherein the peripheral block is one of a phase locked loop and a watchdog timer.

4. (Previously presented) The multithreaded computer based system of claim 1, wherein the embedded pipelined processor further includes an internal memory unit comprising a flash memory with a shadow static memory.

5. (Previously presented) A method of executing instructions in a multithread computer based system having at least a first thread associated with a first context including a set of context registers, the method comprising the steps of:

selecting the first thread associated with the first context;

fetching a first instruction of the first thread which indicates source data registers

associated with operands, each operand associated with a context of one of a

plurality of threads, the context comprising data registers;

decoding the instruction to determine a second context and source data register

associated with a first operand;

executing the instruction on the first operand to produce a result; and

storing the result in a destination data register associated with a third context of one

of the plurality of threads.

6. (Currently amended) The method of claim 5, wherein the decoding further comprises decoding the instruction to determine ~~the~~ a fourth context and ~~the~~ a source data register associated with a second operand, the context associated with the first operand being the ~~first~~ second context and the context associated with the second operand being the ~~second~~ fourth context different from the ~~first~~ second context.

7. (Previously presented) The method of claim 5, wherein the destination data register is part of a second set of context registers of a second thread different from the first thread.

8. (Previously presented) The method of claim 7, wherein the decoding step further comprises determining the third context of the destination data register for storing the result.

9. (Currently amended) The method of claim 5, wherein the executing includes modifying a control and status register to indicate the second context ~~of the first operand~~ being different than the first context.

10. (Previously presented) The method of claim 9, wherein the executing further includes modifying the control and status register to indicate a context of the destination data register being different than the first context.

11. (Previously presented) An apparatus for executing instructions in a multithread computer based system having at least a first thread associated with a first context comprising a set of context registers, the apparatus comprising:

means for selecting the first thread associated with the first context;

means for fetching a first instruction of the first thread which indicates source data

registers associated with operands, each operand associated with a context of

one of a plurality of threads, the context comprising data registers;

means for decoding the instruction to determine a second context and source data

register associated with a first operand;

means for executing the instruction on the first operand to produce a result; and

means for storing the result in a destination data register associated with a third

context of one of the plurality of threads.

12. (Cancelled)

13. (Cancelled)

14. (Cancelled)

15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

18. (Cancelled)

19. (Cancelled)